Tree Generation Algorithms in Computer Graphics

Francisco Peixoto
pg47194@alunos.uminho.pt

***Abstract*—This essay explores the complexity of tree generation algorithms, focusing on traditional methods, L-systems, Tree OL-Systems, and the Space Colonization Algorithm. It emphasizes the role of growth functions in understanding the development of L-systems and the application of the Space Colonization Algorithm in simulating diverse tree forms. The findings highlight the potential and limitations of these algorithms in accurately representing the natural world, underscoring the need for continuous research and development.**

***Index Terms*—Tree Generation Algorithms, L-Systems, Parametric L-Systems, Space Colonization Algorithm, Hierarchical Correspondences, Exploratory Blending, Tree Modeling Techniques, Cross-Species Blending, Axial Trees, Growth Functions**

I. Introduction

In the realm of computer science and mathematics, tree generation algorithms have emerged as a powerful tool for simulating the intricate growth and development of trees. These algorithms, which draw on principles from natural sciences, offer a unique approach to modeling and visualizing complex branching structures. This essay delves into the diverse world of tree generation algorithms, exploring their types, applications, and the underlying principles that guide their operation. From traditional methods to the innovative Space Colonization Algorithm, we will examine how these algorithms capture the essence of nature's complexity. We will also discuss the role of L-systems and Tree OL-Systems in generating self-similar fractals and realistic images of plants. Furthermore, we will explore the significance of growth functions in understanding the development of these systems. Through this exploration, we aim to highlight the potential and limitations of tree generation algorithms in accurately representing the natural world.

II. Understanding the Complexity of Nature

[3] Generally, tree modeling techniques fall into two categories. The first category aims to generate trees that are morphologically similar yet different, while the second category aims to generate inspiring novel trees through cross-species blending.

This dichotomy is beautifully illustrated in the exploration of Salix species, where a series of morphologically similar yet different trees were obtained by blending two species. The same process was used to generate inspiring novel trees by cross-species blending between an artistic Bonsai and a Salix. (Fig. 1.)

One of the key aspects of tree generation algorithms is the establishment of hierarchical fuzzy correspondences. This concept is crucial when the input trees differ significantly in both geometries and topologies. Chains with noticeably different geometries and sub-structures might be matched, leading to the generation of unique and novel tree structures. This process is further enhanced by the use of exploratory N-way blending, where designers can import multiple trees into the system, which spontaneously spans to a blending space. The designers can then hand-pick any interesting point in the space, and the algorithm interactively generates an in-between tree using the parametric coordinates as interpolation weights of the input. (Fig. 2.)

In this context, "Hierarchical Correspondences" refer to the relationships established between different parts of the tree models based on their hierarchical structure. This allows for the matching of different parts of the tree models, even when they have significantly different geometries and topologies.

While "Exploratory Blending" is a method that allows designers to blend multiple tree models together. The system creates a blending space that contains all possible combinations of the input tree models. Designers can then select any point in this space, and the system will generate a new tree model that is a blend of the input models, with the blend weights determined by the selected point's parametric coordinates. This method allows for the creation of a wide variety of unique and novel tree models.

III. Types of Tree Generation Algorithms and Their Applications

A. Axial Trees

Tree generation algorithms can be broadly categorized into several types, each with its unique characteristics and applications. One of the fundamental concepts in tree generation algorithms is the axial tree. As Prusinkiewicz and Lindenmayer (1990) explain,

> Axial trees are purely topological objects. The geometric connotation of such terms as straight segment, lateral segment, and axis should be viewed at this point as an intuitive link between the graph-theoretic formalism and real plant structures.

The ordering of branches in axial trees, introduced by Gravelius and further developed by Horton and Strahler, is crucial in synthesizing botanical trees. This ordering scheme allows for the creation of realistic tree structures that mimic the branching patterns observed in nature. (Fig. 3.)

Fig. 1. A sequence of blended trees. The top row shows the series of morphologically similar yet different trees (middle) obtained by blending between two Salix species (left and right). The bottom row demonstrates our capability of generating inspiring novel trees by cross-species blending between an artistic Bonsai (left) and a Salix (right)

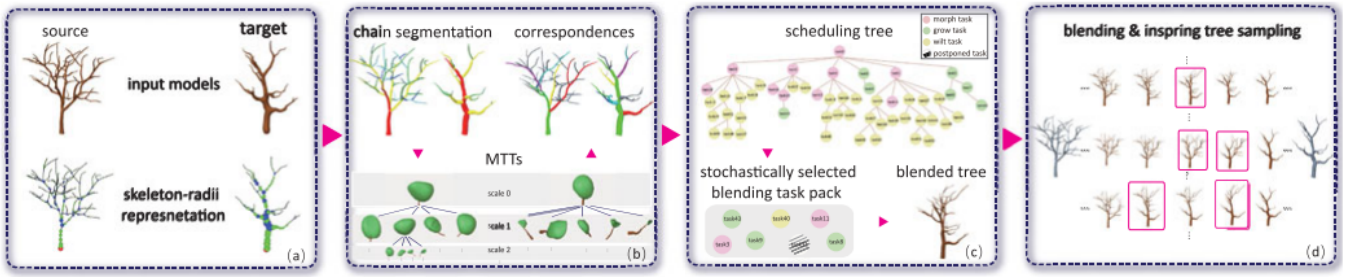


Fig. 2. An overview of this method. Skeleton-radii representations of input trees (a), chain segmentation, multi-scale topology trees construction and fuzzy correspondence establishment (b), stochastic blending task scheduling (c), and multiple batches of blending and the selection of inspiring trees (circled in red)

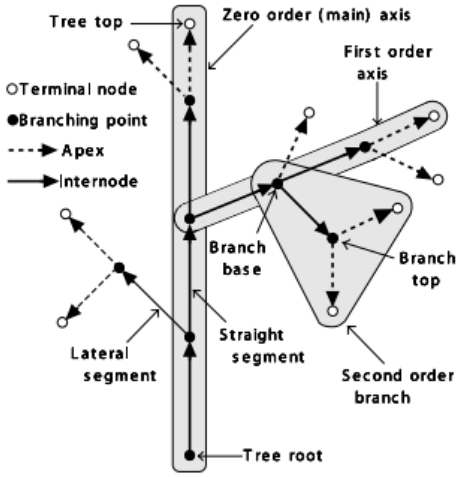


Fig. 3. An axial tree

B. Space Colonization Algorithms

In addition to these traditional tree generation algorithms, a novel approach known as the Space Colonization Algorithm has been developed. This algorithm treats competition for space as the key factor determining the branching structure of trees.

It begins with an initial configuration of attraction points and one or several tree nodes. The tree is generated iteratively, with each iteration potentially influencing the tree node that is closest to an attraction point. This influence occurs if the distance between the point and the closest node is less than a radius of influence. The new nodes are incorporated into the tree structure, extending the main axis and initiating lateral branches.

The algorithm involves repetitively testing the set of attraction points for proximity to the tree nodes, and attraction points are removed when there is at least one tree node closer to them than a threshold kill distance. This method generates a wide variety of trees and shrubs, controlled by a few parameters and algorithm variations.

The models generated with the space colonization algorithm are visually plausible even as bare trees and shrubs, without leaves that could potentially mask shortcomings of the branching structures. In particular, branch intersections are prevented by the nature of the algorithm. When needed, the generated branching structures can be complemented with leaves, flowers, buds, and fruits. (Fig. 4.)

IV. L-Systems and Parametric L-Systems

[2] L-systems, also known as Lindenmayer systems, are a type of formal grammar most famously used to model the growth processes of plant development. They were introduced and developed by Aristid Lindenmayer, a Hungarian theoretical biologist and botanist, in 1968. L-systems have found

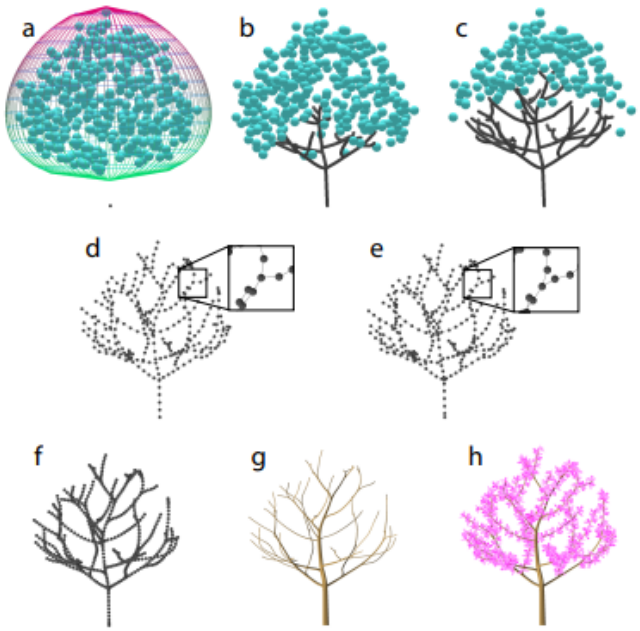
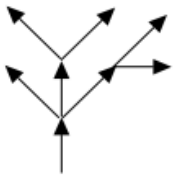


Fig. 4. Key steps of the proposed method. a) Envelope with the attraction points and the initial tree node; b, c) Generation of the tree skeleton; d) Node decimation; e) Node relocation; insets show the modified branching angle; f) Subdivision; g) Construction of generalized cylinders; h) Addition of organs.

applications in the field of computer graphics, where they are used to generate self-similar fractals and realistic images of plants.

An L-system consists of an alphabet of symbols that can be used to make strings. It also consists of a collection of production rules that expand each symbol into some larger string of symbols, an initial axiom string from which to begin construction, and a mechanism for translating the generated strings into geometric structures. L-systems are thus parallel rewriting systems, i.e., all the productions are applied simultaneously in each derivation step. (Fig. 5.)



F[+F][-F[-F]F]F[+F][-F]

Fig. 5. Bracketed string representation of an axial tree.

Parametric L-systems are a variant of L-systems that allow parameters to be attached to symbols in the L-system alphabet. These parameters can be used to control various aspects of the modelled plant, such as the length and angle of branches. The use of parameters allows a more detailed and precise control over the generated structures, making it possible to create more realistic and diverse models.

For example, a context-sensitive production in a parametric L-system might look like this:

$$A(x) < B(y) > C(z) : x + y + z > 10 \rightarrow E\left(\frac{x+y}{2}\right) F\left(\frac{y+z}{2}\right)$$

This production can be applied to a module B(5) that appears in a parametric word, leading to the generation of unique and novel tree structures.

These systems provide a powerful tool for expressing developmental models that involve diffusion of substances throughout an organism, as illustrated by an extended model of the pattern of cells observed in Anabaena catenula and other blue-green bacteria. (Fig. 6.)

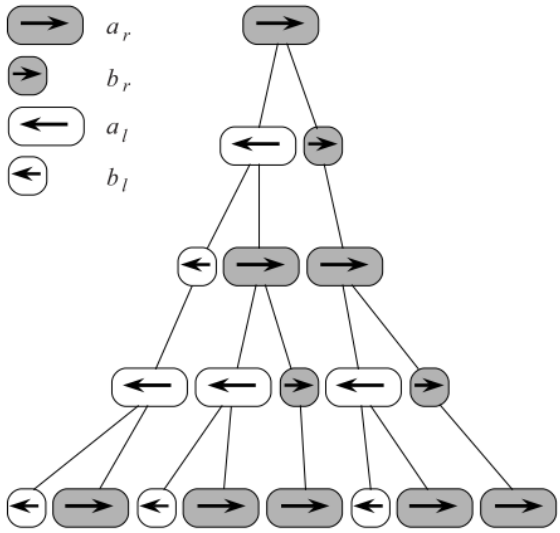


Fig. 6. Development of a filament (Anabaena catenula) simulated using a DOL-system.

Procedural methods play a crucial role in the application of L-systems and parametric L-systems. These methods involve the use of procedures or algorithms as part of the production rules. For example, a procedural method could be used to control the branching angle based on the age of the branch, or to control the thickness of the branch based on its length. The use of procedural methods allows for a high degree of flexibility and control over the generated models, enabling the creation of complex and realistic tree structures.

V. Tree OL-Systems

Tree OL-Systems are a powerful tool for modeling the development of branching structures. These systems operate directly on axial trees, using a rewriting mechanism known as a tree production. This process replaces a predecessor edge with a successor axial tree, aligning the starting node of the predecessor with the base of the successor, and the ending node with the top of the successor.

A tree OL-system is defined by three components: a set of edge labels (V), an initial tree (ω) with labels from V, and a set of tree productions (P). The edge labels provide a way to identify and differentiate between the various edges in the tree. The initial tree serves as the starting point for the development of the branching structure, while the tree productions dictate the rules for how the tree should grow and develop over time.

The application of these tree productions can be repeated iteratively, allowing for the modeling of complex, multi-branched structures. This makes tree OL-systems particularly useful in fields such as computer graphics and biological modeling, where they can be used to generate realistic representations of trees and other branching structures. (Fig. 7.)

A. Iterative Application of Tree Productions

The iterative application of tree productions allows for the modeling of complex, multi-branched structures. This feature of tree OL-systems makes them particularly useful in fields such as computer graphics and biological modeling. They can be used to generate realistic representations of trees and other branching structures, providing a powerful tool for visual and analytical applications.

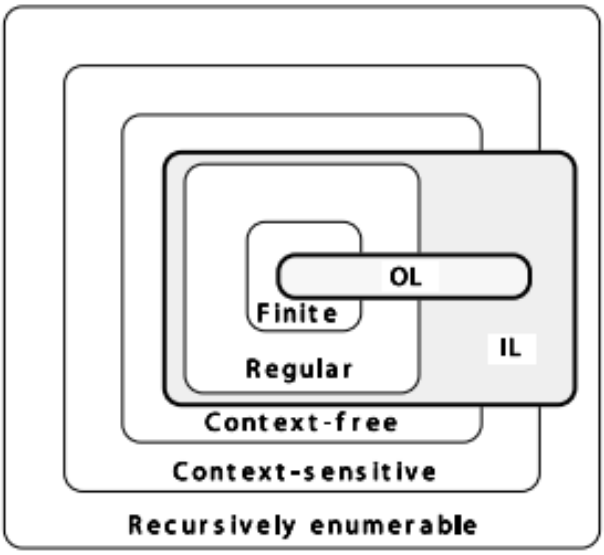


Fig. 7. Relations between Chomsky classes of languages and language classes generated by L-systems. The symbols OL and IL denote language classes generated by context-free and context-sensitive L-systems, respectively.

B. Growth Functions

Growth functions play a vital role in understanding the development of L-systems. They describe the number of symbols in a word based on its derivation length. In the context of plant modeling, growth functions help differentiate between branching patterns and elongation of plant segments. The elongation is often represented by sequences of symbols, with the number of symbols doubling in each derivation step, resulting in exponential growth. The analysis of growth functions in L-systems involves the use of matrix notation to express the relationship between the number of letter occurrences in derived words.

C. Matrix Notation in Growth Functions

To analyze growth functions effectively, the use of matrix notation proves invaluable. Matrix notation enables the comparison of letter occurrences in pairs of words, providing a concise and systematic approach. By constructing a square matrix, where each entry represents the number of occurrences of a letter in the successor of a production, the analysis becomes more structured and manageable. The independence of growth functions in DOL-systems from letter ordering allows for a convenient representation using matrix notation. This notation enhances our understanding of the growth patterns and relationships between different letters within the system.

VI. Space Colonization Algorithm

[1] The Space Colonization Algorithm is a unique approach to tree generation that is fundamentally based on the concept of spatial competition, a key determinant of tree branching structures. The algorithm initiates with a set of spatially distributed attraction points and a singular initial node.

In the iterative process of the algorithm, tree nodes are progressively generated, taking into account the attraction points and a predefined radius of influence. Each node possesses a specific influence area, and when an attraction point is located within this area, it stimulates the growth of a new branch.

The algorithm continues this iterative process until all attraction points have been utilized, or the addition of new nodes is no longer possible. The outcome is a tree structure that effectively colonizes the available space, reflecting the natural competition for space observed in tree growth.

The operation of the algorithm begins with an initial configuration of N attraction points and one or several tree nodes. The tree is generated iteratively. In each iteration, an attraction point may influence the tree node that is closest to it, if the distance between the point and the closest node is less than a radius of influence di.

A new tree node v' will be created and attached to v by segment (v, v') if there are attraction points that influence a single tree node v. The node v' is positioned at a distance D from v, in the direction defined as the average of the normalized vectors toward all the sources $s \in S(v)$. The direction of growth can be biased by a vector g representing a combined effect of branch weight and tropisms.

Once the new nodes have been added, a check is performed to test which, if any, of the attraction points should be removed due to the proximity of tree branches that have grown toward these points. An attraction point s is removed when there is at least one tree node v closer to s than a threshold kill distance dk. (Fig. 8.)

The Space Colonization Algorithm is particularly useful in simulating irregular forms of temperate-climate broad-leaved trees. These forms are difficult to capture with older modeling methods, which emphasize recursive aspects of tree structure. The models generated with the space colonization

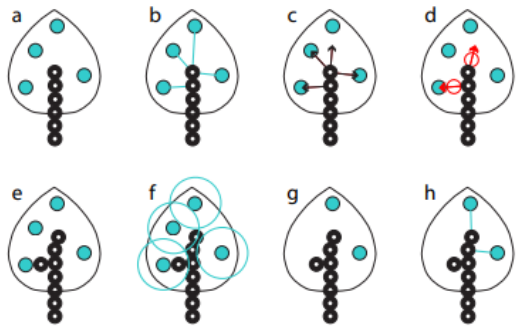


Fig. 8. The space colonization algorithm.

algorithm are visually plausible even as bare trees and shrubs, without leaves that could potentially mask shortcomings of the branching structures.

The Space Colonization Method is efficient, but it's not without its flaws. The time it takes to generate a tree strongly depends on model parameters, especially the segment size D used as the unit of length in the models. At present, there is no algorithmic criterion for choosing the optimal value of D.

A. Biological Justification of the Space Colonization Algorithm

The Space Colonization Algorithm is biologically justifiable. In nature, the competition for space, likely mediated by the quantity and quality of light, has a significant impact on plant form. This competition appears to play the dominant role, overriding other factors in determining the overall branching structure of temperate-climate trees and shrubs.

In forest ecosystems, for instance, trees and plants strive to reach the sunlight, leading to the development of complex branching structures. This natural process is mirrored in the Space Colonization Algorithm, which starts with a set of attraction points distributed in space and a single initial node. The algorithm then iteratively generates the tree nodes based on these attraction points and a defined radius of influence. Each node has a certain area of influence, and if an attraction point falls within this area, it contributes to the growth of a new branch. This process continues until all attraction points are exhausted, or no new nodes can be added, resulting in a tree structure that effectively colonizes the available space.

This algorithm is particularly useful in simulating the growth and development of trees and other branching structures in various fields, including computer graphics, biological modeling, and architectural design. It provides a realistic and efficient method for generating complex structures that mimic the natural growth patterns of plants.

However, it's important to note that while the Space Colonization Algorithm is biologically justifiable, it is still a simplification of the complex processes that occur in nature. Real-world plant growth is influenced by a multitude of factors, including genetic factors, environmental conditions, and interactions with other organisms, which are not fully captured by the algorithm. Therefore, while the Space Colonization Algorithm provides a useful tool for modeling and simulating plant growth, it should be used in conjunction with other methods and tools to achieve a more comprehensive understanding of plant growth and development.

VII. Conclusion

In conclusion, the intricate and multifaceted nature of simulating natural phenomena is beautifully captured in the complexity and diversity of tree generation algorithms. From traditional methods to novel approaches like the Space Colonization Algorithm, L-systems, and Tree OL-Systems, these algorithms provide powerful tools for understanding and visualizing the growth and development of trees. Each algorithm, with its unique characteristics and applications, contributes to our ability to accurately represent the natural world in computer graphics. The study of growth functions and the application of the Space Colonization Algorithm further illustrate the potential of these methods in capturing the complexity and beauty of nature. However, as with any technology, these methods are not without their limitations. Ongoing research and development are essential to refine these tools, expand their capabilities, and explore new applications. As we continue to bridge the gap between computer science, mathematics, and natural sciences, the potential for innovation and discovery is limitless.

References

[1] Brendan Lane Adam Runions and Przemyslaw Prusinkiewicz. Modeling trees with a space colonization algorithm. 2007.

[2] Przemyslaw Prusinkiewicz Aristid Lindenmayer. *The Algorithmic Beauty of Plants*. 1990.

[3] Xiaogang Jin Yutong Wang, Xiaowei Xue. Creative virtual tree modeling through hierarchical topology-preserving blending. *IEEE Transactions on Visualization and Computer Graphics*, 23(12), 2023.